

Evaluating Reliability of Motion Features in Surveillance Videos

Longin Jan Latecki¹, Roland Mieziako¹, Dragoljub Pokrajac²

¹Temple University, CIS Dept., Philadelphia, PA, latecki@temple.edu, rmiezian@temple.edu

²Delaware State University, CIS Dept., Dover, DE, dpokraja@desu.edu

ABSTRACT

Although a tremendous effort has been made to perform a reliable analysis of images and videos in the past fifty years, the reality is that one cannot rely 100% on the analysis results. The only exception is applications in controlled environments as dealt in machine vision, where closed world assumptions apply. However, in general, one has to deal with an open world, which means that content of images may significantly change, and it seems impossible to predict all possible changes. For example, in the context of surveillance videos, the light conditions may suddenly fluctuate in parts of images only, video compression or transmission artifacts may occur, a wind may cause a stationary camera to tremble, and so on. The problem is that video analysis has to be performed in order to detect content changes, but such analysis may be unreliable due to the changes, and thus fail to detect the changes and lead to “vicious cycle”.

The solution pursuit in this paper is to monitor the reliability of the computed features by analyzing their general properties. We consider statistical properties of feature value distributions as well as temporal properties. Our main strategy is to estimate the feature properties when the features are reliable computed, so that any set of features that does not have these properties is detected as being unreliable. This way we do not perform any direct content analysis, but instead perform analysis of feature properties related to their reliability.

The main effort in video analysis nowadays is still in making the feature computation more reliable. Our claim is that we need to accept the fact that the computed features will never be 100% reliable, and focus our attention on computing reliability measures. This way system decisions will only be made when features are sufficiently reliable. This means for an intelligent system for video analysis that in addition to feature computation, it should perform instantaneous evaluation of their reliability, and adapt its behavior in accordance to the reliability. For example, if the goal of a system is to monitor motion activity, and to signal an alarm if the activity is high, the system is allowed to make reliable decisions only if there exist a subset of the computed motion activity features that is sufficiently reliable. The monitoring of features reliability and adjusting the system behavior accordingly, seems to be the only way to deploy autonomous video surveillance systems.

1. INTRODUCTION

We begin by showing two examples of video content changes that cause the existing motion detection approaches to inaccurately detect the presence of substantial motion. Clearly, the detected motion is present in videos, but it is due to some content artifacts and is not due to the actual presence of moving objects. Consequently, human observant ignores such “motion” as irrelevant, while standard video analysis systems detect it as significant activity. We will show that the feature reliability methods proposed in this paper allow us to identify the unreliable motion features, and to ignore the irrelevant artifacts. This is possible without reducing the detection rate of real moving objects. Consequently, we eliminate false alarms without reducing the detection rate. We stress that this is obtained without any direct video content analysis (e.g., using different features), but by monitoring the reliability of computed features. As stated in the introduction, direct video content analysis with further features does not solve the problem, since these features may also become unreliable.

Our first example illustrates motion artifacts in *Campus 3*¹ video introduced by some reflections in windows that are probably caused by cars passing by. In Fig. 1, we show two frames from Campus 3 video, one showing real motion, and the second showing the motion artifacts in addition to the normal motion. Our second example, in Fig. 2(a), shows motion artifacts introduced by video compression. The same scene without such artifacts is shown in Fig. 2(b). This video, which we call *Temple 2*, was recorded in real-world environment by the video surveillance system of the Campus Police Division of the Temple University.

In Section 2, we first describe a simple temporal method to determine the reliability of motion detection. In Section 3, we present a more sophisticated statistical method based on distribution analysis of feature values and information theory [21]. Both methods monitor features computed by our motion detection approach presented in [17], which we summarize in Section 4. The motion features are computed for gray level or infrared videos using 3D spatiotemporal blocks of spatial size 8x8 pixels, and temporal size of 3 frames. The blocks are disjoint in space and overlap by one frame in time. As result we obtain *motion activity* values for each 8x8 block in every video frame. By thresholding the motion activity values, we

¹ *Campus 3* can be obtained from the Performance Evaluation of Tracking and Surveillance (PETS) repository:
ftp://pets.rdg.ac.uk/PETS2002/DATASET1/TESTING/CAMERA3_JPEGS/

obtain a binary feature, called *motion detection*, with 1 standing for ‘motion detected’ and 0 for ‘no motion detected’. Both videos as well as our motion detection results can be viewed on [12].

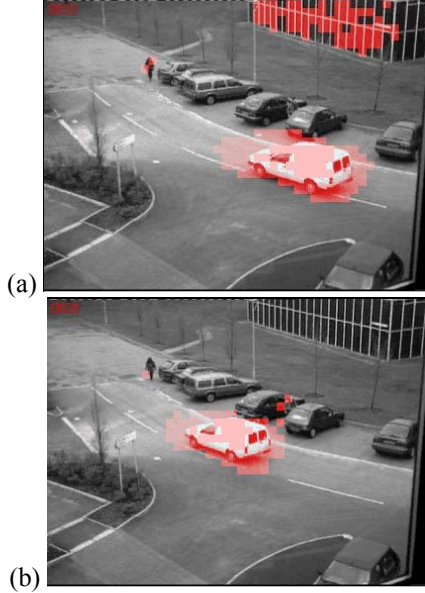


Figure 1. Two frames from *Campus 3* video with moving blocks highlighted red: (a) motion artifacts due to reflections in the windows, (b) the same scene (a few frames later) without the artifacts.

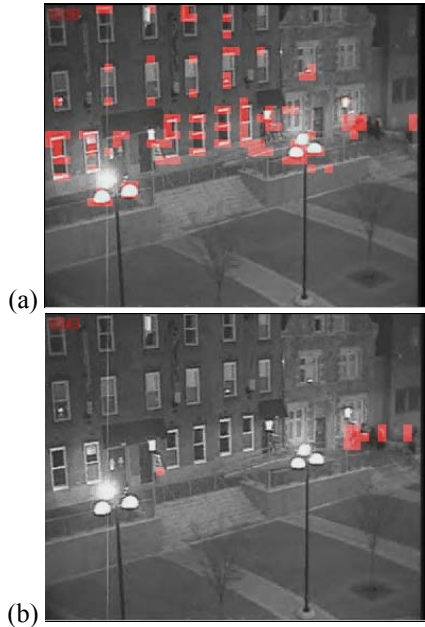


Figure 2. Two frames from *Temple 2* video with moving blocks highlighted red: (a) motion artifacts introduced by video compression, (b) the same scene (a few frames later) without the artifacts.

A temporal reliability analysis introduced in Section 2 is applied to the motion detection feature, while a statistical reliability analysis introduced in Section 3 is applied to the motion activity feature.

2. TEMPORAL ANALYSIS OF FEATURE RELIABILITY

In this section, we describe a simple temporal method to determine the reliability of motion features. The input motion feature has binary values for each 8×8 block for each video frame with 1 for ‘motion detected’ and 0 for ‘no motion detected’. The algorithm described in Section 4.2 computes this feature vector. The 8×8 blocks are disjoint. Let $f(n)$ be the number of 1s in the frame number n , i.e., $f(n)$ is the number of detected moving blocks as function of frame number. We use the finite difference approximation of first derivative of f to monitor the reliability of our motion detection. In simple words, if the jump in values of f is above a certain threshold for a given time interval, the binary feature is unreliable in this interval. The threshold necessary to detect the unreliable features is not static. We propose a dynamic thresholding algorithm described in Section 2.1 to learn and vary this threshold. However, some other learning techniques could also be used.

This reliability property works under the assumption that there exists an upper bound on the size of moving objects whose motion we want to detect (measured in the number of moving blocks). This assumption holds for most surveillance videos. Now we consider an example video, called *Temple 2*, that satisfies this assumption. This video is recorded by a roof mounted, stationary camera, so that a certain minimal distance to moving objects is guaranteed. Typical moving objects there, humans and vehicles, cannot get arbitrarily large. Hence, the fraction of the scene occupied by a moving object is limited. Observe that the actual value of the upper bound on the size of moving objects needs not to be known, since our algorithm learns it automatically.

In Fig. 3(a), we see the graph of function f for *Temple 2* video. Time intervals with significant jumps of f that are correctly identified by our dynamic thresholding are marked with red lines in Fig. 3(b). The graph of modified feature f , when f was set to zero within the time intervals when motion detection was detected as unreliable is shown in Fig. 3(c). Fig. 3(c) shows that the proposed method is able to identify and exclude the unreliable results of motion detection. By excluding these time intervals from further processing, we not only eliminate false alarms, but make possible to correctly detect alarm situations, although the input motion detection is not 100% reliable. For example, a significant increase in the number of motion blocks after the frame 1700 indicates an alarm situation. This is a correct prediction, since a street fight is recorded on the video after the frame 1700, see the *Temple 2* video [12].

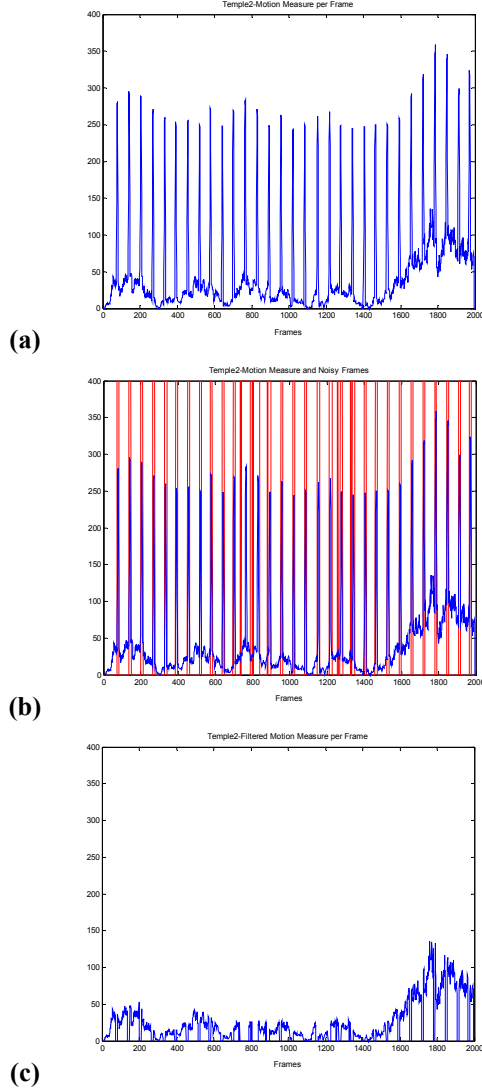


Figure 3. (a) The graph of $f(n)$, which is the number of moving blocks as function of frame number n . (b) Significant jumps of f (caused by feature unreliability) correctly identified by our dynamic thresholding. (c) The graph of f padded by zeros for frames with unreliable motion detection.

2.1. DYNAMIC THRESHOLDING ALGORITHM

Now we describe a dynamic thresholding algorithm used to detect the jumps of function f . First we compute the initial values of mean $meanl$ and standard deviation $stdl$ using all previous values of $f(x)$ for $x=1, \dots, t-1$ and some time instance t . The actual dynamic thresholding starts at time $x=t$. A jump up is detected at points $x \in \{x+1, x+2, \dots, x+w\}$ for a window size w if

$$meanrw(x) - meanl(x) > C1 * stdl(x),$$

where $C1$ is a constant. A dynamic threshold values $meanl$ and $stdl$ are updated if

$$meanrw(x) - meanl(x) < C2 * stdl(x),$$

where $C2 < C1$ is a second constant. The updated values are:

$$meanl(x) = u * meanl(x) + (1-u) * meanrw(x)$$

$$stdl(x) = u * stdl(x) + (1-u) * stdrw(x)$$

where u is a learning constant and

$$meanrw(x) = \frac{1}{w} \sum_{\tau=1}^w f(x+\tau)$$

$$stdrw(x) = \sqrt{\frac{1}{w-1} \sum_{\tau=1}^w (d(x+\tau) - meanrw(x))^2}.$$

The symmetric window constant w was set to 3, giving us a sliding window of 7 frames ($2*w+1$). The learning constant was $u=0.9$. Constants $C1$, $C2$ of function f used in detecting jumps of the *Temple 2* video were selected based on the initial running average $meanl$ and $stdl$. The value of $meanl$ was 10.3 and the value of $stdl$ was 7.4. Constant $C1$ was set to 15 and constant $C2$ was set to 3 providing the initial jump detected threshold to 154.5 and reset to no-jump detected threshold of 30.9.

3. STATISTICAL ANALYSIS OF FEATURE RELIABILITY

To determine whether a particular feature is reliable, we assume that the feature bears more information if its distribution differs more significantly from a normal (Gaussian) distribution. Similar heuristics are used e.g., in Independent Components Analysis [20]. The follow-up assumption is that the feature becomes unreliable if an addition random noise is superimposed, which would lead the distribution of such noisy feature to become more Gaussian like. Hence, by measuring to what extent a feature distribution differs from a Gaussian distribution, one can not only get information to what extent the feature is useful but also when such usefulness drops (e.g., due to some external and often non-observed factor).

The Information Theory proposes *negentropy* as the measure of this discrepancy. Given a probability density $p(x)$ of a feature, Differential Entropy is defined [18, 19] as:

$$H(x) = \int_{-\infty}^{\infty} -p(x) \log p(x) dx \quad (1)$$

For a given class of distributions $p(x)$ that have the same variance σ^2 , differential entropy is maximal for a Gaussian distribution where it is equal

$$H_{Gauss}(\sigma^2) = \frac{1}{2} \log 2\pi e \sigma^2. \quad (2)$$

Hence, a negentropy, which defined as

$$J(x) = H_{Gauss}(\sigma^2) - H(x) \quad (3)$$

or its normalized value

$$J(x)/H_{Gauss}(\sigma^2) = 1 - \frac{H(x)}{H_{Gauss}(\sigma^2)} \quad (4)$$

may be used to measure usefulness and reliability of a feature. Observe that the minimal value of negentropy is 0 (when $p(x)$ is Gaussian).

A naïve approach to compute negentropy would be to employ histograms to approximate $p(x)$ with piecewise linear function $p'(x)$ such that:

$$p'(x) = Kp(x_i), x \in [x_i, x_i + \Delta x]$$

where K is a normalization constant (chosen such that $p'(x)$ is a distribution). However, as shown in [21] this non-parametric technique is very unstable since dependent on a proper choice of a histogram bin size Δx and histogram centers x_i . Hence we use parametric approach suggested in Hyvarinen's NIPS 1997 paper [18]. The main ideas of this approach are:

1) Instead of original feature x , use a standardized feature $x^* = (x - \text{mean}(x)) / \text{std}(x)$ that have zero mean and unit standard deviation. This way, we could directly use negentropy to compare reliability with no need to normalize with the entropy of a Gaussian.

2) Use a first-order Taylor approximation of a logarithmic function in eq. (1) that leads to: $(1 + \varepsilon) \log(1 + \varepsilon) \approx \varepsilon + \varepsilon^2 / 2$;

3) Use conveniently chosen set of orthogonal functions of $G_i(x)$ of a feature x to expand probability density function $p(x)$ in vicinity of a Gaussian probability density.

In practice, the choice of orthogonal functions is based on practicability and sensitivity on outliers of the computation of estimates for expectations $E(G_i(x))$, integrability of the obtained probability density function approximation and last, but not the least, the properties of non-Gaussian distributions we want to capture.

Based on such consideration, [18] proposes the following two approximations of negentropy, that we use in this paper:

$$J_a(x^*) = k_1 E(x^* \cdot e^{-x^{*2}/2})^2 + k_{2a} \left(E(|x^*|) - \sqrt{\frac{2}{\pi}} \right)^2 \quad (5a)$$

$$J_b(x^*) = k_1 E(x^* \cdot e^{-x^{*2}/2})^2 + k_{2b} \left(E(e^{-x^{*2}/2}) - \sqrt{\frac{1}{2}} \right)^2 \quad (5b)$$

where the coefficients are determined as:

$$k_1 = \frac{36}{8\sqrt{3}-9}, k_{2a} = \frac{24}{2-\frac{6}{\pi}}, k_{2b} = \frac{24}{16\sqrt{3}-27} \quad (6)$$

The proposed technique is applicable on any continuous feature. In this paper, we evaluate the reliability of the *motion activity* feature, defined in [17] (see Section 4) as the largest eigenvalue of texture vectors in a small time window. For each frame, we standardize the feature values x^* , compute expectations $E(|x^*|)$, $E(x^* \cdot e^{-x^{*2}/2})$ and finally compute the negentropy approximations eq. (5a), (5b) per frame.

We evaluated the proposed techniques for assessing feature reliability on a set of videos [12]. This set includes infrared videos, for which the same settings of parameters as for visual light videos were used. Here we focus on our results on two video sequences from the Performance Evaluation of

Tracking and Surveillance (PETS) repository: a sequence from PETS2001² here referred to as *Campus 1* sequence, a sequence from PETS2002, here referred to as the *Campus 3* sequence and on a *Temple 2* sequence from Temple University.

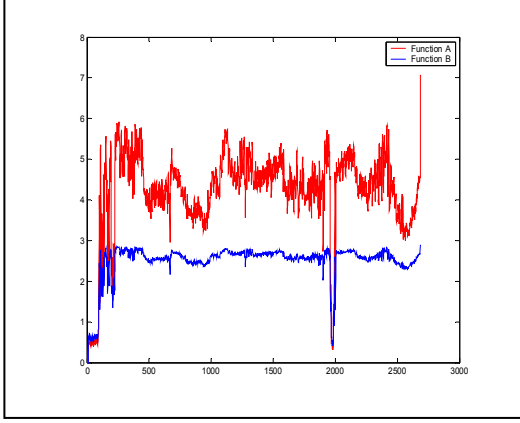
Campus 1. At the beginning of the sequence, there is no movement, so changes in the motion activity (an observed feature) are random, which reflects small negentropy values in approximately first 100 frames, see Fig 4(a). Both negentropy approximations (eq. 5a, 5b) demonstrate strong drop between frames 1960 and 2000 which corresponds to the higher level of noise that can be visually observed between these frames. Function B (eq. 5b) provides more stable approximation values, which makes it potentially more useful.

Campus 3. Both methods identified drop around frames 330, 660, a strong drop around 700, a drop around 720 and the relatively long-term drop between 800 and 900, see Fig. 4(b). Finally, there were some small oscillations between 1200 and 1300 and one drop around 1400. All these events correspond to frames in the video sequence when our algorithm has difficulties in properly identifying moving objects based on observed feature (e.g., due to reflections in the upper right part of the frame, cp. Fig. 1). Again Function B (eq. 5b) performed better, by having less oscillations and fluctuations.

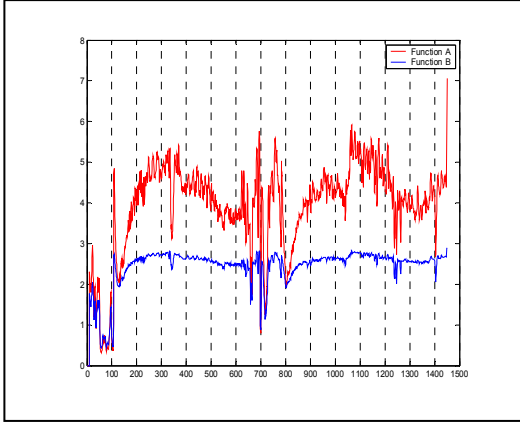
Temple 2. On this video, there is evident instability (manifested as flicker) that can be traced to applied compression technique. The period of this disturbance, which has negative effects on motion detection, is around 62 frames. Using the proposed technique, we obtained negentropy values that reflect this periodicity. Both functions eq. (5a) and (5b) have strong periodical components, see Fig. 4(c), and demonstrate oscillations which period can be correctly determined using a Fast Fourier Transform [22], as approximately 62 frames. Function (5b) is again more stable and provides better automatic period estimation. The results of the statistical method agree with those of the temporal methods, cp. Fig. 3.

A common denominator of the results shown is that the proposed negentropy-based technique can help in determining frames when the observed feature is unreliable (periodic or pulse flicker, noise, etc.). Since both eq. 5a and 5b are only relatively rough approximations of negentropy, there is no wonder they do not provide the same values, especially when a negentropy is relatively high. As expected, when a negentropy is low, the feature probability distribution is closer to a Gaussian so both approximations would give similar results. Generally, eq. (5b) provides better performance. It is more stable and has less fluctuations. Hence is potentially more suitable for automatic thresholding.

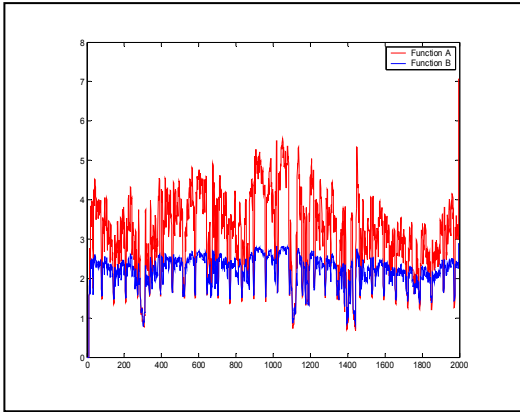
²http://pets.rdg.ac.uk/PETS2001/DATASET1/TESTING/CAMERA1_JPEGs/



(a)



(b)



(c)

Figure 4. Estimated negentropy per frame of each video using (eq. 5a) in red and (eq. 5b) in blue for (a) *Campus 1*; (b) *Campus 3*; (c) *Temple 2* videos.

4. FEATURE GENERATION AND MOTION DETECTION

We shortly describe our motion detection method proposed in [17]. It is based on change analysis of texture vectors computed for 3D, spatiotemporal (sp) blocks. In our previous paper [11] we have shown that the use of sp texture vectors of 3D blocks in the framework of Stauffer and Grimson [14] can improve the detection of moving objects while potentially cutting back the processing time due to the reduction of the number of input vectors per frame. Our experimental results in [17] (videos be viewed on [12]) show that our motion detection technique leads to further performance improvements.

4.1 Video representation with spatiotemporal (sp) texture vectors

We represent videos as three-dimensional (3D) arrays of gray level or monochromatic infrared pixel values $g_{i,j,t}$ at a time instant t and a pixel location i, j . A video is characterized by temporal dimension Z corresponding to the number of frames, and by two spatial dimensions, characterizing number of pixels in horizontal and vertical direction of each frame.

We divide each image in a video sequence into disjoint $N_{BLOCK} \times N_{BLOCK}$ squares (e.g., 8×8 squares) that cover the whole image. Spatiotemporal (3D) blocks are obtained by combining squares in consecutive frames at the same video plane location. In our experiments, we used $8 \times 8 \times 3$ blocks that are disjoint in space but overlap in time, i.e., two blocks at the same spatial location at times t and $t+1$ have one square in common.

The fact that the 3D blocks overlap in time allows us to perform successful motion detection in videos with very low time frequency, e.g., in our experimental results [12] videos with 2 fps (frame per second) are included. The obtained 3D blocks are represented as 192-dimensional vectors of gray level or monochromatic infrared pixel values. We then zero mean these vectors and project them to three dimensions using principal component analysis (PCA). The obtained 3-dimensional vectors form a compact spatiotemporal texture representation for each block. The PCA projection matrices are computed separately for all video plane locations (a set of disjoint 8×8 squares in our experiments).

The blocks are represented by N -dimensional vectors $\mathbf{b}_{I,J,t}$, specified by spatial indexes (I, J) and time instant t . Vectors $\mathbf{b}_{I,J,t}$ contain all values $g_{i,j,t}$ of pixels in the corresponding 3D block.

To reduce dimensionality of $\mathbf{b}_{I,J,t}$ while preserving information to the maximal possible extent, we compute a projection of the normalized block vector to a vector of a significantly lower length $K \ll N$ using a PCA projection matrix $\mathbf{P}_{I,J}^K$ computed for all $\mathbf{b}_{I,J,t}$ at video plane location (I, J) . The resulting sp texture vectors $\mathbf{b}_{I,J,t}^* = \mathbf{P}_{I,J}^K \mathbf{b}_{I,J,t}$ provide a joint representation of texture and motion patterns in videos

and are used as input of algorithms for detection of moving objects. We used $K=3$ in our experiments.

To compute $\mathbf{P}_{I,J}^K$ we employ the principal values decomposition following [4,5]. A matrix of all normalized block vectors $\mathbf{b}_{I,J,t}$ at video plane location (I,J) is used to compute the $N \times N$ dimensional covariance matrix $\mathbf{S}_{I,J}$. The PCA projection matrix $\mathbf{P}_{I,J}$ for spatial location (I,J) is computed from the $\mathbf{S}_{I,J}$ covariance matrix. The projection matrix $\mathbf{P}_{I,J}$ of size $N \times N$ represents N principal components. By taking only the principal components that corresponds to the K largest eigenvalues, we obtain $\mathbf{P}_{I,J}^K$.

4.2 Moving objects detection based on local variation

The assumption of the proposed technique is that the variation of location vectors—corresponding to the same location within a small number of consecutive frames— will increase if the vectors correspond to a moving object. In practice, for each location (x,y) , we consider vectors

$$\mathbf{b}_{x,y,t-W}^*, \mathbf{b}_{x,y,t-W+1}^*, \dots, \mathbf{b}_{x,y,t}^*, \dots, \mathbf{b}_{x,y,t+W}^*$$

corresponding to a symmetric window of size $2W+1$ around the temporal instant t . For these vectors, we compute the covariance matrix $\mathbf{C}_{x,y,t}$. We assign the largest eigenvalue of

$\mathbf{C}_{x,y,t}$, denoted as $\lambda_{x,y,t}$, to a given spatiotemporal video position to define a local *variance measure*, which we will also refer to as *motion activity*

$$ma(x,y,t) = \lambda_{x,y,t}.$$

The larger the variance measure $ma(x,y,t)$, the more likely is the presence of a moving object at position (x,y,t) . Finally, we label each video position as moving or stationary (background) depending whether the motion activity is larger or smaller than a suitably defined threshold. We use a dynamic thresholding algorithm (described in Section 2) to determine the threshold value at position (x,y,t) based on the history of $ma(x,y,s)$ values over time ($s=1, \dots, t-1$).

5. CONCLUSIONS

In this paper, we proposed and evaluated two methods to monitor the reliability of features applied in video surveillance and motion detection. The methods have been evaluated on real-life surveillance videos. Both methods correctly identified time intervals when an observed feature becomes non useful for motion detection (e.g., due to flicker, artifacts introduced by compression algorithm, etc.). The proposed methodology is potentially applicable to other domains where unsupervised learning is performed under open-world assumption (where we cannot anticipate all the events which could occur during the operational life of an automated intelligent system).

6. ACKNOWLEDGEMENTS

D. Pokrajac has been partially supported by NIH-funded Delaware Biomedical Research Infrastructure Network

(BRIN) Grant (P20 RR16472), and DoD HBCU/MI Infrastructure Support Program (45395-MA-ISP Department of Army).

7. REFERENCES

- [1] Buttler, D., Sridharan, S., and Bove, V. M. Real-time adaptive background segmentation. In *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, Baltimore 2003.
- [2] R.T. Collins, A.J. Lipton, and T. Kanade, "Introduction to the Special Section on Video Surveillance", *IEEE PAMI* 22(8) (2000), pp. 745–746.
- [3] Devore, J. L., *Probability and Statistics for Engineering and the Sciences*, 5th edn., Int. Thomson Publishing Company, Belmont, 2000.
- [4] Duda, R., P. Hart, and D. Stork, *Pattern Classification*, 2nd edn., John Wiley & Sons, 2001.
- [5] Flury, B. A *First Course in Multivariate Statistics*, Springer Verlag, 1997.
- [6] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-Time Surveillance of People and Their Activities", *IEEE PAMI* 22(8) (2000), pp. 809–830.
- [7] Jain, R., Militzer, D., and Nagel, H. Separating nonstationary from stationary scene components in a sequence of real world TV images. In *Proc. IJCAI*, 612–618, Cambridge, MA, 1977
- [8] Jolliffe, I. T, *Principal Component Analysis*, 2nd edn., Springer Verlag, 2002.
- [9] Javed, O., Shafique, K., and Shah, M. A. Hierarchical approach to robust background subtraction using color and gradient information. In *Proc. IEEE Workshop on Motion and Video Computing (MOTION)*, 22-27, Orlando, 2002,.
- [10] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions", *IEEE PAMI* 22(8) (2000), pp. 831–843.
- [11] D. Pokrajac and L. J. Latecki: Spatiotemporal Blocks-Based Moving Objects Identification and Tracking, *IEEE Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, October 2003.
- [12] R. Mieziako, L. J. Latecki, D. Pokrajac. Link to test results. <http://knight.cis.temple.edu/~video/VA>
- [13] Remagnino, P., G. A. Jones, N. Paragios, and C. S. Regazzoni, eds., *Video-Based Surveillance Systems*, Kluwer Academic Publishers, 2002.
- [14] C. Stauffer, W. E. L. Grimson, "Learning patterns of activity using real-time tracking", *IEEE PAMI* 22(8) (2000), pp. 747–757.
- [15] Westwater, R., Furht, B., *Real-Time Video Compression: Techniques and Algorithms*, Kluwer Academic Publishers, 1997.
- [16] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfindex: Real-time Tracking of the Human Body", *IEEE PAMI* 19(7) (1997), pp. 780–785.
- [17] L. J. Latecki, R. Mieziako, and D. Pokrajac. Motion Detection Based on Local Variation of Spatiotemporal Texture. *CVPR Workshop on Object Tracking and*

Classification Beyond the Visible Spectrum (OTCBVS), Washington, July 2004.

[18] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems*, volume 10, pages 273-279. MIT Press, 1998.

[19] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[20] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

[21] D. Pokrajac and L. J. Latecki. Entropy-Based Approach for Detecting Feature Reliability. *Invited Paper, 48th Conf. for Electronics, Telecommunications, Computers, Automation, and Nuclear Engineering (ETRAN)*. Cacak, Serbia, June 2004.

[22] E. Oran Brigham. *The Fast Fourier Transform: An Introduction to Its Theory and Application*. Prentice Hall, 1973.